L Number	Hits	Search Text	DB	Time stamp
1	0	prioritiz$4 near4 (lock$3 adj2 way$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:40
2	0	prioritiz$4 near8 (lock$3 adj2 way$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:41
3	0	prioritiz$4 near8 (lock$3 near3 way$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:41
4	0	prioritiz$4 same (lock$3 near3 way$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:41
5	0	prioritiz$4 same (lock$3 near8 way$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:42
6	32602	lock$3 near8 way$2	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:42
7	4	prioritiz$4 same (LRU near2 way$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:47
8	4	prioritiz$4 same LRU same lock$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:49
9	16	prioritiz$4 near5 LRU	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:49
10	1126745	lock$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:50

11	5	(prioritiz$4 near5 LRU) and lock$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 15:55
12	19	(multilevel or (multi adj level)) adj2 inclusion	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 16:19
13	390	first adj bus adj interface	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:04
14	358	second adj bus adj interface	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 16:20
15	284	(first adj bus adj interface) and (second adj bus adj interface)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 16:20
16	1980	snoop$ near4 cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:09
18	0	((multilevel or (multi adj level)) adj2 inclusion) and (((first adj bus adj interface) and (second adj bus adj interface)) and (snoop$ near4 cache))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 16:21
17	26	((first adj bus adj interface) and (second adj bus adj interface)) and (snoop$ near4 cache)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 16:24
19	2635014	power	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 16:24
20	11	(((first adj bus adj interface) and (second adj bus adj interface)) and (snoop$ near4 cache)) and power	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:15

21	52	low adj power adj bus	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:07
22	29	high adj power adj bus	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:07
23	9	(low adj power adj bus) and (high adj power adj bus)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:08
24	4719	snoop$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:08
25	2	((low adj power adj bus) and (high adj power adj bus)) and snoop$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:08
26	61157	cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:09
27	2	((low adj power adj bus) and (high adj power adj bus)) and cache	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:09
28	112333	low adj power	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:15
29	3	(((first adj bus adj interface) and (second adj bus adj interface)) and (snoop$ near4 cache)) and (low adj power)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:53
30	34541	clock adj generator	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:05

31	23	((first adj bus adj interface) and (second adj bus adj interface)) and (clock adj generator)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:54
32	2	(((first adj bus adj interface) and (second adj bus adj interface)) and (clock adj generator)) and (snoop$ near4 cache)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:54
33	12	((first adj bus adj interface) and (second adj bus adj interface)) and (low adj power)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 17:55
34	7	(((first adj bus adj interface) and (second adj bus adj interface)) and (low adj power)) and snoop$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:03
35	2767	low adj power adj mode	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:03
36	2	(low adj power adj mode) and ((first adj bus adj interface) and (second adj bus adj interface))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:04
37	37	(low adj power adj mode) and (snoop$ near4 cache)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:04
38	7433	first adj bus	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:08
39	25	((low adj power adj mode) and (snoop$ near4 cache)) and (first adj bus)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:05
40	3	(clock adj generator) and (((low adj power adj mode) and (snoop$ near4 cache)) and (first adj bus))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:16

41	7811	second adj bus	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:08
42	1	((clock adj generator) and (((low adj power adj mode) and (snoop$ near4 cache)) and (first adj bus))) and (second adj bus)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:08
43	3	((clock adj generator) and (((low adj power adj mode) and (snoop$ near4 cache)) and (first adj bus))) and snoop$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/11/23 18:16

US-PAT-NO: 6480965

DOCUMENT-IDENTIFIER: US 6480965 B1
See image for Certificate of Correction

TITLE: Power management method for a computer system having a
hub interface architecture

DATE-ISSUED: November 12, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Harriman; David J.	Sacramento	CA	N/A	N/A
Poisner; David I.	Folsom	CA	N/A	N/A
Rabe; Jeff	Rancho Cordova	CA	N/A	N/A

US-CL-CURRENT: 713/322

ABSTRACT:

According to one embodiment, a computer system includes a Central Processing
Unit (CPU), a hub agent and a hub interface coupled to the first hub agent.
The computer system transitions from a first power state to a second power
state upon the CPU determining that no requests are pending to access the first
hub interface.

20 Claims, 12 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 12

---------- KWIC ---------

Brief Summary Text - BSTX (8):

According to one embodiment, a method of reducing power of a computer system
including a hub interface architecture is disclosed. The method includes
operating in a first power state and transitioning to a second power state upon
detecting that no requests are pending to access a **first bus**.

Drawing Description Text - DRTX (7):
FIG. 5 is a flow diagram of one embodiment for the transition of a computer
system from a **low power mode** to a 1.times.mode;

Drawing Description Text - DRTX (8):
FIG. 6 is a flow diagram of one embodiment for the transition of a computer
system from a **low power mode** to a 4.times.mode;

Detailed Description Text - DETX (16):
Referring back to FIG. 2, ICH 140 is coupled to a **clock generator** 280.
Clock generator 280 generates synchronizing clock pulses that provide the
fundamental timing and internal operating frequency for components within
computer system 100. In one embodiment, ICH 140 is the clock controlling agent
within computer system 100 that controls the operation of **clock generator** 280
and other synchronization devices within system 100.

Detailed Description Text - DETX (18):
One drawback to operating hub interfaces in the 4.times.mode is that a
higher quantity of power is consumed by computer system 100. Further, although
less power is consumed during the 1.times.mode, significant amounts of energy
may be wasted in the event one or more of the hub interfaces is not used while
system 100 is maintained in a powered-up state. This can be particularly
disadvantageous for portable computers, which rely on batteries for their
power. Therefore, in order to make computer system 100 more energy-efficient,
the hub interfaces operate in a clock disabled (**low power) mode**. In the **low**
power mode, the system clock and PLLs are turned off and interface operation is
suspended to further lower power consumption in computer system 100.

Detailed Description Text - DETX (19):
FIG. 4 is a flow diagram for one embodiment of the sequence for
transitioning from the 1.times.or 4.times.modes to the **low power mode** in
computer system 100. At process block 410, CPU 410 determines that no requests

are pending to access hub interface A. At process block 420, interface masters 362 are disabled to prevent access of hub interface A from being granted to either MCH 110 or ICH 140, except for the power management sequence messages described below. At process block 430, ICH 140 begins the **low power mode** sequence by asserting a signal that puts CPU 102 in a powered down state. In response, CPU 102 transmits an acknowledge signal that is propagated over hub interface A to ICH 140. At process block 440, ICH 140 broadcasts to all hub interface agents connected to ICH 140 that system computer 100 is to enter the **low power mode**. For example, signals are transmitted to MCH 110, network interface 160 and bridge 165 indicating that the power down mode will soon be entered. Further, hub interface agents coupled to ICH 140 that are also coupled to a downstream agent transmit the **low power mode** signal to the downstream agent.

Detailed Description Text - DETX (20):

At process block 450, ICH 140 receives acknowledge signals from all of the hub interface agents indicating that the agents are ready to enter the **low power mode**. At process block 460, ICH 140 stops **clock generator** 280. Also, all agent PLLs are stopped if system 100 is operating in the 4.times.mode, or some other mode requiring a PLL (e.g., 8.times.mode). According to one embodiment, **clock generator** 280 and the PLLs are stopped by gating their respective outputs. At process block 470, all hub interface agents enter the **low power mode**. Computer system 100 remains in the **low power mode** until ICH 140 receives a request indicating a need to exit the **low power mode**.

Detailed Description Text - DETX (21):

FIG. 5 is a flow diagram for one embodiment of the operations for the transition of computer system 100 from a **low power mode** to a 1.times.mode. At process block 510, an asynchronous signal is received at ICH 140 via RQA signal 254. Since system 100 is in the **low power mode**, ICH 140 interprets the received asynchronous signal as indicating a need to exit the **low power mode**.

For example, MCH 110 may transmit the asynchronous signal to ICH 140 if it is necessary for graphics interface 113 to access may memory 115. In other embodiments, MCH 110 may be coupled to another hub interface agent via a hub interface. In such an embodiment, an asynchronous signal may be transmitted to MCH 110 if the agent needs to exit the **low power mode**. The signal is subsequently transmitted to ICH 140 via MCH 140.

Detailed Description Text - DETX (22):

At process block 520, ICH 140 transmits the asynchronous signal to other hub agents in system 100 via the RQB signal 255 on each hub interface bus. For instance, after receiving the signal from MCH 110, ICH 140 relays the signal to network interface 160 and bridge 165 via hub interface busses B and C, respectively. The asynchronous signals may be regenerated at pulse shaper 368 before being transmitted to the other hub agents via selector 364. At process block 530, ICH 140 enables **clock generator** 368. At process block 540, ICH 140 enables CPU 102 by deasserting the signal that shut down CPU 102 in the power down sequence. At process block 550, all hub interface agents begin operating the 1.times.mode.

Detailed Description Text - DETX (23):

FIG. 6 is a flow diagram for one embodiment of the operations for the transition of computer system 100 from a **low power mode** to a 4.times.mode. At process block 605, a triggering event occurs requiring computer system 100 to wake up (e.g., a user striking a key on a keyboard in computer system 100). At process block 610, an asynchronous signal is received at ICH 140 as RQA signal 254. At process block 620, ICH 140 transmits the asynchronous signal to other hub agents in system 100 as a RQA signal 254 on each hub interface bus.

Detailed Description Text - DETX (24):

At process block 630, ICH 140 enables **clock generator** 368. In addition, PLLs within each hub agent are enabled. Since each agent includes its own PLL, it may be necessary to synchronize the PLLs on each side of a hub interface.

Therefore, at process block 640, the upstream agent on each hub interface
transmits a small packet of data to the downstream agent via data path 251
after it wakes up. Since there is a delay between the time in which ICH 140
enables **clock generator** 368 and the PLLs and their actual activation, the
packet of data is clocked to the downstream agents by STROBE signals 258 using
source synchronous data transfers. The packet of data is stored in the
downstream agents in register A of registers 366. The downstream agent may or
may not be awake upon receiving the packet of data since the clocks may not yet
be activated.

Detailed Description Text - DETX (46):
In alternative embodiments, additional transaction attributes may include
the ability to differentiate between **"snooped" traffic where cache** coherency is
enforced by hardware (i.e., chipset) and "non-snooped" traffic that relies on
software mechanisms to ensure data coherency in the system. Moreover, another
possible attribute would be an "explicitly prefetchable" hint, to support a
form of read caching and allow for more efficient use of the main memory
bandwidth.

Claims Text - CLTX (2):
2. The method of claim 1 further comprising: disabling masters within first
and second agents coupled to the first hub interface after detecting that no
requests are pending to access the first hub interface; placing a central
processing unit (CPU) in a low powered state; and gating a **clock generator**
coupled to the first agent.

Claims Text - CLTX (4):
4. The method of claim 3 further comprising: receiving an asynchronous
signal at the first agent indicating that the computer system is to transition
back to the first power state; ungating the **clock generator**; placing the CPU
in a high power state; and transitioning to the first power state.

Claims Text - CLTX (7):
7. The method of claim 6 further comprising: receiving an asynchronous signal from the first agent indicating that the computer system is to transition back to the first power state; ungating the **clock generator**; ungating the PLLs within the first and second; agents placing the CPU in a high power state; and transitioning to the first power state.

Claims Text - CLTX (8):
8. The method of claim 7 further comprising: transmitting a packet of data from the second agent to the first agent via the first hub interface after starting the **clock generator** and the PLLs; and transmitting an acknowlege signal from the first agent to the second agent.

Claims Text - CLTX (13):
13. The computer system of claim 12 further comprising a **clock generator** coupled to the first hub agent, wherein the interface master, the CPU and the **clock generator** are disabled upon detecting that no requests are pending to access the first hub interface.

US-PAT-NO: 5813022

DOCUMENT-IDENTIFIER: US 5813022 A

TITLE: Circuit for placing a cache memory into **low power mode** in response to special bus cycles executed on the bus

DATE-ISSUED: September 22, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ramsey; Jens K.	Houston	TX	N/A	N/A
Stevens; Jeffrey C.	Spring	TX	N/A	N/A
Tubbs; Michael E.	Montgomery	TX	N/A	N/A
Stancil; Charles J.	Tomball	TX	N/A	N/A

US-CL-CURRENT: 711/3, 365/189.07 , 365/227 , 365/230.03 , 711/118 , 711/143 , 711/146 , 711/167 , 713/320 , 713/323

ABSTRACT:

A circuit for placing an external or L2 cache memory into **low power mode** in response to certain special cycles executed by the microprocessor. In particular, the special cycles are the stop grant acknowledge special cycle and the halt special cycle. The microprocessor executes the stop grant acknowledge special cycle in response to a request by the computer system to slow down its clock. This request is asserted by the computer system if the system has been idle for a predetermined period of time. The halt special cycle is generated by the microprocessor when a HALT instruction is executed. The stop grant acknowledge and halt special cycles place the microprocessor into a low power state. Since the microprocessor is in **low power mode,** the L2 cache memory is also placed into **low power mode** for further power conservation. The L2 cache memory is implemented either with synchronous or asynchronous static random access memories (SRAMs). To place a synchronous SRAM into **low power**

mode, its
address strobe input is asserted while its chip select input is deasserted.
For an asynchronous SRAM, deasserting its chip select input causes the SRAM to
transition into **low power mode**.

9 Claims, 10 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 10

---------- KWIC ---------

Abstract Text - ABTX (1):
A circuit for placing an external or L2 cache memory into **low power mode** in
response to certain special cycles executed by the microprocessor. In particular, the special cycles are the stop grant acknowledge special cycle and
the halt special cycle. The microprocessor executes the stop grant acknowledge
special cycle in response to a request by the computer system to slow down its
clock. This request is asserted by the computer system if the system has been
idle for a predetermined period of time. The halt special cycle is generated
by the microprocessor when a HALT instruction is executed. The stop grant
acknowledge and halt special cycles place the microprocessor into a low power
state. Since the microprocessor is in **low power mode,** the L2 cache memory is
also placed into **low power mode** for further power conservation. The L2 cache
memory is implemented either with synchronous or asynchronous static random
access memories (SRAMs). To place a synchronous SRAM into **low power mode,** its
address strobe input is asserted while its chip select input is deasserted.
For an asynchronous SRAM, deasserting its chip select input causes the SRAM to
transition into **low power mode**.

TITLE - TI (1):
Circuit for placing a cache memory into **low power mode** in response to
special bus cycles executed on the bus

Brief Summary Text - BSTX (3):
The invention relates to a technique for conserving power in computer systems, and more particularly, to circuitry that responds to certain special bus cycles by placing cache memory devices in the computer system into **low power mode**.

Brief Summary Text - BSTX (6):
However, even with the use of CMOS technology, power dissipation by the computer system components still causes significant amounts of energy to be wasted, particularly when the computer system is powered on but not in use. Heretofore, efforts have concentrated on stopping or slowing down the clock to the main microprocessor and turning off peripheral devices such as disk drives, video systems and I/O ports to conserve power. Such an apparatus for reducing computer system power consumption is described in U.S. Pat. No. 4,980,836, entitled "Apparatus for Reducing Computer System Power Consumption," by Carter, et al. Carter describes a computer system that monitors its address bus to determine when selected peripheral devices have not been accessed for a preset amount of time. When the preset amount of time has expired, the computer system powers itself down and stops the system clock. In addition, the keyboard electronics are placed into **low power mode** and power is removed from various miscellaneous logic. However, the memory devices remained powered up to retain programs. Later advances allowed powering off even the memory devices by storing the entire system state to hard disk before completely powering down. But this is a final step and power conservation is desired at all levels and periods, not just after very long periods.

Brief Summary Text - BSTX (10):
A circuit according to the present invention responds to certain computer special cycles by placing the cache memory system into **low power mode**. In particular, the preferred embodiment responds to a stop grant acknowledge special cycle and a halt special cycle. The stop grant special cycle is

preferably asserted by the microprocessor when a signal, referred to as a stop clock signal, indicating a request to slow down or stop the computer system clock signal, is asserted. The stop clock signal is asserted whenever the system detects that it has been idle for a predetermined period of time. The halt special cycle is preferably generated by the microprocessor in response to a halt instruction executed by the microprocessor, which causes the microprocessor to cease operation and enter a low power state. In both such cycles, the operation of the microprocessor has either been slowed down or stopped. As a result, the cache memory system is not in use and thus is placed into **low power mode** to conserve energy. Another feature of the present invention is that the microprocessor and cache memory system are temporarily taken out of **low power mode** to perform certain cache consistency cycles and placed back into **low power mode** after the cycles have completed.

Detailed Description Text - DETX (8):

Referring now to FIG. 1, the system board S of an exemplary computer system incorporating the preferred embodiment of the present invention is shown. In the preferred embodiment, the system board S contains circuitry and slots for receiving interchangeable circuit boards. In the preferred embodiment, there are two primary buses located on the system board S. The **first bus** is the PCI or Peripheral Component Interconnect bus 98 which includes address/data portion 100, also referred to as PCIAD, control and byte enable portion 102 and control signal portion 104. The address/data bus PCIAD is preferably 32 bits wide, although it can be upgraded to 64 bits if desired. The second primary bus on the system board S is the EISA bus 99. The EISA bus 99 includes LA address portion 106, SA address portion 108, SD data portion 110 and EISA/ISA control signal portion 112. The PCI and EISA buses 98 and 99 form the backbones of the system board S.

Detailed Description Text - DETX (16):

The microprocessor 200 includes an internal or L1 cache memory. A level 2

(L2) or external cache memory system 208 is connected to the processor bus 202
to provide additional caching capabilities to improve performance of the
computer system. A processor cache and memory controller (CMC) and PCI bridge
chip 210 is connected to the control portion PC and to the address portion PA.
The CMC 210 receives the signal SLEEP* provided by the miscellaneous logic chip
132. to place the microprocessor 200 into **low power mode**. The CMC 210 also
controls data buffers 212 and 213. Additionally, the CMC 210 provides control
signals MC and address signals MA to a main memory 214. The control signals MC
include various RAS* and CAS* signals. The CMC 210 is also connected to the L2
cache memory 208, as it incorporates the cache controller, and therefore,
controls the operations of cache memory devices 209 in the L2 cache system 208.
The cache memory devices 209 are implemented with static random access memories
(SRAMs). For the 32-bit processors, cache memory sizes of 128K or 256K bytes
are supported, with the cache line width being 16 bytes. For the 64-bit
processors, cache sizes of 256K and 512K bytes are supported, with the cache
line width being 32 bytes.

Detailed Description Text - DETX (26):
In the preferred embodiment, the L2 cache memory 208, whether implemented
with asynchronous or synchronous cache SRAMs 209, is placed into **low power mode**
whenever a stop grant acknowledge cycle is detected on the processor bus 202 or
a HLT command is executed by the microprocessor 200. Upon execution of a HLT
instruction, the microprocessor 200 generates a halt special cycle on the
processor bus 202. The halt special cycle is indicated by the following bus
definition signals: byte enable bit PBE[2]* is asserted low, signals M/IO* and
D/C* are asserted low, a signal W/R* is driven high, and the address bits
PA[31:0] are driven to the value 0.times.00000000. The stop grant acknowledge
special bus cycle is driven by the microprocessor 200 in response to its
STPCLK* input being asserted low, which is a request to slow down or stop the

clock input of the microprocessor 200. The signal STPCLK* is asserted by the
CMC 210 in response to the assertion of the signal SLEEP* by the miscellaneous
logic chip 132. When the signal STPCLK* is asserted, the microprocessor 200
stops execution on the next instruction boundary unless superseded by a higher
priority interrupt. Then the microprocessor 200 generates a stop grant
acknowledge special bus cycle. The definition of the stop grant acknowledge
bus cycle is identical to the halt special cycle except that the address bits
PA[31:0] are driven to the value 0.times.00000010. Thus, a halt special cycle
and a stop grant acknowledge special cycle differ only in address bit PA[4].
For a description of the signals discussed above, refer to IntelDX4 Processor
Data Book (1994) (hereinafter "IntelDX4.TM. Data Book"); and Pentium Processor
User's Manual. Volume 1: Pentium Processor Data Book (1994) (hereinafter
"Pentium.RTM. Data Book"), which are hereby incorporated by reference.

Detailed Description Text - DETX (27):
Synchronous or burst SRAMs include the following inputs: address strobe
inputs ADSP* and ADSC* and chip select inputs CS*. The synchronous SRAM is
placed into **low power mode** by asserting its address strobe input ADSC* low but
maintaining its chip select input CS* deactivated. Burst cycles are initiated
in the synchronous SRAMs 209 by asserting either their ADSP* or ADSC* input
pins, which cause the initial burst address to be strobed into the SRAMs. The
ADSP* inputs of the cache SRAMs 209 are connected to the processor address
strobe signal ADS* driven by the microprocessor 200 and the ADSC* inputs are
connected to address strobe signals CADS0* or CADS1* provided by the CMC 210.
Asynchronous SRAMs do not include address strobe inputs ADSC* or ADSP*.

Instead, they have an address latch input CALEN for latching in the external
address. Asynchronous SRAMs are placed into **low power mode** simply by
deasserting their chip select inputs CS*.

Detailed Description Text - DETX (33):
The D flip flop 314 is clocked by the signal CLK2, and its D input is

connected to the output of the AND gate 312. The inputs of the AND gate 312
are connected to the inverted state of the address strobe ADS* and the output
of an OR gate 316. The inputs of the OR gate 316 receive signals PAHOLD.sub.--
DLY and the inverted state of a signal PAHOLD. Assertion of the signal PAHOLD
causes the CMC 210 to assert an address hold signal AHOLD to the microprocessor
200, which causes the microprocessor 200 to tristate all its address signal
outputs. The signal AHOLD is driven high to allow the CMC 210 to drive a **snoop**
address onto the address bus PA during a **snoop** cycle, which allows the
microprocessor 200 and the CMC 210 to determine if the addressed data is stored
in the L1 cache or the L2 cache memory 208. For non-**snoop** cycles, the signal
PAHOLD is deasserted low, thereby enabling the AND gate 312 to detect
activation of the address strobe ADS*. The signal PAHOLD.sub.-- DLY is the
signal PAHOLD delayed by one CLK2 cycle, and is used to qualify the address
strobe ADS* to determine if an L1 writeback cycle initiated by the
microprocessor 200 is occurring. An L1 writeback cycle occurs if a hit occurs
to a modified line in the L1 cache of the microprocessor 200. Even though the
signal AHOLD is maintained asserted by the CMC 210, the microprocessor 200
nevertheless asserts the address strobe ADS* to initiate the L1 writeback
cycle. Thus, in an L1 writeback cycle, detection of the address strobe ADS*
occurs one CLK2 cycle after AHOLD is asserted high.

Detailed Description Text - DETX (35):
A state machine GREENST responds to assertion of the signal GREEN.sub.-- REQ
by asserting various signals to indicate **low power mode**. On reset, indicated
by a signal RESET being asserted high, the state machine GREENST enters into
state A, where it remains until the signal GREEN.sub.-- REQ is asserted high.
The state machine GREENST provides output signals D.sub.-- GREENMODE, D.sub.--
GREEN.sub.-- PAHOLD and D.sub.-- GREEN.sub.-- CADS, which are all deasserted
low in state A. Unless otherwise indicated, output signals of state machines
are assumed to be deasserted. On the assertion of the signal GREEN.sub.-- REQ,
control proceeds from state A to state B. The signals D.sub.--

GREENMODE and
D.sub.-- GREEN.sub.-- PAHOLD are asserted high in the transition. The signal
D.sub.-- GREENMODE is provided to the D input of a D flip flop 324, which is
clocked by the signal CLK2 and reset low on the rising edge of the signal
RESET. The D flip flop 324 provides a signal GREENMODE which, when asserted
high, indicates to the rest of CMC 210 that **low power mode** is currently active.
The signal D.sub.-- GREEN.sub.-- PAHOLD is provided to an input of an OR gate
326. The OR gate 326 also receives a signal D.sub.-- MYSNP.sub.-- PAHOLD
asserted during **snoop** cycles. The other inputs of the OR gate 326 receive
other signals asserted during other cycles. The output of the OR gate 326 is
connected to the D input of a D flip flop 328, which is clocked by the signal
CLK2, and whose output is connected to the input of a tristate buffer 329. The
tristate buffer is enabled by the active low signal IOBUFOE.sub.--, and it
drives the address hold signal AHOLD. The signal IOBUFOE.sub.-- is normally
asserted low and is deasserted high only during certain test cycles. The
signal AHOLD is asserted until the CMC 210 returns the ready acknowledge signal
RDY* or BRDY* to the microprocessor 200. The signal AHOLD is asserted here to
prevent the microprocessor 200 from possibly starting another cycle while the
cache SRAMs 209 are in the process of being powered down.

Detailed Description Text - DETX (43):
Returning now to the state machine GREENST, the assertion of the signals
SYNC.sub.-- ENDPCYC and T2 causes control to transition from state B to state
C, where the state machine remains until a signal LEAVE.sub.-- GREEN is
asserted high. In state C, the output signal D.sub.-- GREENMODE is maintained
high to indicate **low power mode**. The signal LEAVE.sub.-- GREEN is provided by
an AND gate 322, which receives the inverted state of the address strobe ADS*
and a signal HITM* provided by the microprocessor 200 to indicate a hit to a
modified line in the L1 **cache during a snoop** cycle. When the address strobe
ADS* is asserted by the microprocessor 200 in conjunction with the assertion of

the signal HITM*, a writeback cycle of the modified line to main memory 214 is
indicated. Since the microprocessor 200 is able to respond to a **snoop** cycle
while it is in the stop grant state or the auto halt power down state, the
writeback of a modified line does not require that the microprocessor 200 be
taken out of **low power mode**. As a result, the state machine GREENST stays in
state C and the signal D.sub.-- GREENMODE is maintained asserted high. The
signal LEAVE.sub.-- GREEN is not driven high until ADS* is detected in a
non-writeback cycle. It is also noted that the signal HITM* is provided only
by the Pentium processors. When the CMC 210 is used with the 486 DX4
processor, a pullup resistor 338 is provided to pull the signal HITM* high.
Assertion of the signal LEAVE.sub.-- GREEN causes the state machine GREENST to
transition from state C back to state A and to deassert the signal D.sub.--
GREENMODE.

Detailed Description Text - DETX (55):
Thus, if asynchronous SRAMs 209 are used, and the signal D.sub.-- GREENMODE
is asserted high to indicate **low power mode**, the AND gate 508 drives its output
high if the processor bus 202 is inactive and a writeback cycle is not in
progress. As will be described later, this condition causes the chip select
CS* inputs of the asynchronous cache SRAMs 209 to be deasserted high to place
the SRAMs into **low power mode**. For an asynchronous SRAM, its chip select input
CS* must be maintained high for it to stay in lower power mode. A synchronous
SRAM is different, however, since placing it into lower power mode involves
asserting its address strobe input ADSC* while maintaining its chip select
input CS* deasserted. Thereafter, the chip select CS* input can be driven low
again without taking the synchronous SRAM out of **low power mode**. The
synchronous SRAM comes out of **low power mode** only if one of its address strobe
inputs ADSP* or ADSC* is activated.

Detailed Description Text - DETX (67):
In summary, if the L2 cache memory 208 is implemented with synchronous

SRAMS, and the signal D.sub.-- GREEN.sub.-- CADS is asserted high by the state machine GREENST in response to a halt special bus cycle or a stop grant acknowledge bus cycle, address strobe signals CADS0* and CADS1* are asserted low by the CMC 210. The signal D.sub.-- GREEN.sub.-- CADS also controls chip select signals CCS0* and CCS1*. If the L2 cache memory 208 is configured as direct-mapped or two-bank two-way set associative, then both signals CCS0* and CCS1* are deasserted high; however, in the case of the direct-mapped configuration, the signal CCS1* is ignored. If the L2 cache memory 208 is configured as single-bank two-way set associative, then the signal CCS1* is utilized effectively as an address signal to select between way 0 and way 1, and thus does not play a role in placing the cache SRAMs 209 into low power state. In single-bank two-way set associative mode, assertion of the address strobes CADS0* and CADS1* and deassertion of the chip select signal CCS0* places the cache SRAMs 209 into **low power mode**. This is accomplished by driving the signal D.sub.-- GREEN.sub.-- CADS high for one CLK2 cycle. Once in **low power mode,** the synchronous cache SRAMs 209 remain there until the processor address strobe ADS* or the cache address strobes CADSn* are asserted low.

Detailed Description Text - DETX (68):

For asynchronous cache SRAMs 209, as discussed above, the AND gate 508 forces the chip select signals CCSn* to be deasserted high while the signal D.sub.-- GREENMODE is active to maintain the cache SRAMs 209 in **low power mode**.

Detailed Description Text - DETX (69):

In the preferred embodiment, two conditions exist that may cause the cache SRAMs 209 to temporarily transition out of low power state. The first involves a **snoop** hit to a modified line in the internal or L1 cache of the microprocessor 200. This can only occur with the Pentium P54 or P24 processors, as they support the MESI writeback protocol. The 486 DX4 microprocessor, on the other hand, supports only the writethrough protocol. In response to the **snoop** hit to the modified line, the Pentium microprocessor 200 asserts its HITM* output. Assertion of the signal HITM* informs the

CMC 210
that a **snoop** or inquire cycle has hit a modified line in the internal cache of
the microprocessor 200, and that the microprocessor 200 will write back the
modified line to the main memory 214. For more information on inquire cycles,
refer to the Pentium Data Book.

Detailed Description Text - DETX (70):
If the **snoop** cycle occurred as a result of a memory read cycle initiated by
a PCI bus master, then the L2 cache memory 208 is updated if the address
provided during the writeback cycle matches an entry in the L2 cache memory
208. Updating the L2 cache memory 208 requires taking the cache SRAMs out of
low power mode. The Pentium processor indicates a burst writeback cycle by
asserting a cacheability signal CACHE* in a write cycle. It is noted that
writeback cycles are burst cycles because the line width of the L1 cache is 32
bytes while the data bus PD width is 8 bytes if a 64-bit processor is used.
For a 32-bit processor, the L1 cache line width is 16 bytes and the data bus PD
width is 4 bytes. For burst transfers, four data transfers are expected by the
Pentium processor, wherein completion of each data transfer is indicated by the
assertion of the signal BRDY* by the CMC 210. Thus, the signal HITM* is
maintained asserted low by the microprocessor 200 until the fourth BRDY* is
returned by the CMC 210.

Detailed Description Text - DETX (71):
The second condition that would cause the cache SRAMs 209 to come out of low
power state temporarily is if a **snoop** hit occurs to a modified line in the L2
memory cache system 208. In this instance, the modified line from the L2 cache
memory 208 must be written back to main memory 214. To perform the burst
writeback cycle, a read of the cache SRAMs 209 is performed to retrieve the
modified line.

Detailed Description Text - DETX (72):
Referring now to FIGS. 6A and 6B, logic is shown that handles the

burst
writeback cycle executed by the microprocessor 200 in response to a **snoop** hit
to a modified line in the L1 cache. Portions of a state machine LOCWRST for
detecting the burst writeback cycle are described. On system reset, the state
machine LOCWRST enters state A, where it remains until a signal BWRT2 is
asserted high and a signal MWR.sub.-- REQ or a signal WRHIT.sub.-- REQ is
asserted high. The signal BWRT2 is provided by an AND gate 602 and indicates
that a burst write transfer on the processor bus 202 is occurring. The first
input of the AND gate 602 is connected to the output of a latch 604, and second
input of the AND gate 602 receives the signal T2. The input of the latch 604
is connected to the output of an AND gate 606. The output state of the AND
gate 606 is latched by the latch 604 when the signal PTRK.sub.-- A falls low,
which occurs when an active cycle is detected on the processor bus 202. The
inputs of the AND gate 606 receive the signal W/R* and the inverted state of
the signal CACHE*. When driven high, the signal W/R* indicates a write cycle.
If the signal CACHE* is also asserted low, then a burst writeback cycle is
indicated.

Detailed Description Text - DETX (76):
The AND gate 622 receives signals NEWCYC.sub.-- VALID, POSSIBLE.sub.--
BURST.sub.-- HIT and T.sub.-- MATCHA. The signal POSSIBLE.sub.--
BURST.sub.--
HIT is asserted high if the CMC 210 detects a burst writeback cycle, the L2
cache 208 is enabled, and the snoop request to the L1 cache in the
microprocessor 200 is in response to a memory read operation initiated by a PCI
bus master. The signal T.sub.-- MATCHA is asserted if a match occurs between
the label portion of the address bus PA and the label stored in the selected
entry of the tag RAM 230, regardless of whether the entry is valid or not.
Thus, if the CMC 210 detects a burst writeback cycle on the processor bus 202,
the writeback address matches an entry in the tag RAM 230, the tag RAM 230 is
selected, and the **snoop** request is a result of a memory read cycle by a
PCI bus

master, the AND gate 622 drives its output high, which causes the OR gate 614
to drive the signal WRHIT.sub.-- REQ high.

Detailed Description Text - DETX (90):
After assertion of four BRDY* signals, the signal BCNTF is asserted high,
causing the state machine LOCWRST to transition from state E to state A. In the
transition, the state machine LOCWRST deasserts the signal LOCWR.sub.--
HOLD.sub.-- CA to allow a new address to flow into asynchronous cache SRAMs 209
from the processor address bus PA. In addition, if the signal GREENMODE is
asserted high to indicate that the microprocessor 200 is in either the stop
grant or auto halt power down states, the state machine LOCWRST asserts the
signal D.sub.-- LOCWR.sub.-- GREEN.sub.-- CADS high. The signal D.sub.--
LOCWR.sub.-- GREEN.sub.-- CADS is provided to OR gates 502, 504 and 506 (FIG.
5) to assert signals SYNC.sub.-- CADS, SYNC.sub.-- BLK.sub.-- CCS0 and
SYNC.sub.-- BLK.sub.-- CCS1, respectively. Thus, assertion of the signal
D.sub.-- LOCWR.sub.-- GREEN.sub.-- CADS places the cache SRAMs 209 back into
low power mode.

Detailed Description Text - DETX (92):
Referring now to FIGS. 7A and 7B, logic is shown for responding to a **snoop**
hit to a modified line in the L2 cache memory 208 when the cache SRAMs 209 are
in **low power mode**. A state machine WBST controls the retrieval of a line of
data from the L2 cache memory 208 to write back to the main memory 214. On
system reset, the state machine WBST enters into state A, where it remains
until a signal L2SNP.sub.-- WBREQ is asserted high. The signal L2SNP.sub.--
WBREQ indicates that a hit to a modified line in the L2 cache 208 has occurred
in response to a memory cycle initiated by a PCI bus master. This causes the
state machine WBST to transition from state A to state G. A signal D.sub.--
SEL.sub.-- WBA.sub.-- TO.sub.-- PAOUT is asserted high to select the writeback
address to drive to the processor address bus PA. The writeback address is a
latched version of the PCI address associated with the PCI memory request.

Detailed Description Text - DETX (93):
During the **snoop** cycle, a signal D.sub.-- MYSNP.sub.-- PAHOLD is asserted high in the CMC 210. This signal is provided to an input of the OR gate 326 (FIG. 3A) to assert the address hold signal AHOLD. Asserting the address hold signal AHOLD causes the microprocessor 200 to float its address output pins so that the CMC 210 can drive the processor address bus PA with the writeback address.

Detailed Description Text - DETX (94):
The state machine WBST remains in state G if a signal Q.sub.-- P2MWBFULL is asserted high, which indicates that the processor-to-memory queue is not available for the current writeback cycle. In state G, the output signal L2WB.sub.-- HOLD.sub.-- CA is asserted high to latch the writeback address on the processor address bus PA into cache SRAMs 209, if asynchronous SRAMs are used. The signal D.sub.-- SEL.sub.-- WBA.sub.-- TO.sub.-- PAOUT is also maintained high in state G. Once the processor-to-memory queue becomes available for a writeback request, that is, the signal Q.sub.-- P2MWBFULL is negated, the state machine WBST transitions from state G to state B. The signal L2WB.sub.-- HOLD.sub.-- CA is maintained high to keep the writeback address latched in the SRAMs 209, and the signal D.sub.-- SEL.sub.-- WBA.sub.-- PAOUT is maintained high to continue selecting the writeback address. For synchronous cache SRAMs 209, the signal D.sub.-- L2WB.sub.-- CADS is asserted high to strobe the initial burst address into the L2 cache memory 208. The signal D.sub.-- L2WB.sub.-- CADS is provided to an input of the OR gate 502, which controls the state of the CADSn* signals. Thus, assertion of the signal D.sub.-- L2WB.sub.-- CADS causes the cache SRAMS 209 to come out of **low power mode**.

Detailed Description Text - DETX (99):
If the L2 cache memory 208 is configured as two-bank two-way associative, both chip select signals CCS0* and CCS1* are maintained low to enable

both
banks. However, only the outputs of the selected bank of the L2 cache SRAMs
209 are driven. This is accomplished through the use of signal D.sub.--
L2WB.sub.-- COE0 and D.sub.-- L2WB.sub.-- COE1, respectively. The signal
D.sub.-- L2WB.sub.-- COE0 is equated with the inverted state of the signal
COE.sub.-- WAY, and the signal D.sub.-- L2WB.sub.-- COE1 which is equated to
the state of the signal COE.sub.-- WAY. The signal COE.sub.-- WAY is provided
by an AND gate 712, whose inputs receive the signals L2WB.sub.-- WAY and
TWOBANK.sub.-- 2WAY. As noted above, the signal L2WB.sub.-- WAY when asserted
high indicates a hit to a valid line in the tag RAM 232 during a **snoop** cycle.
Thus, the signal D.sub.-- L2WB.sub.-- COE0 is asserted high for a hit to the
first way and the signal D.sub.-- L2WB.sub.-- COE1 is asserted high for a hit
to the second way. The signal D.sub.-- L2WB.sub.-- COE0 is provided to an
input of an OR gate 714, and the signal D.sub.-- L2WB.sub.-- COE1 is provided
to an input of an OR gate 716. The outputs of the OR gates 714 and 716 are
connected to the D inputs of D flip-flops 718 and 720, respectively. The OR
gates 714 and 716 also receive other signals during other cycles requiring a
read of the L2 cache memory 208. Both D flip-flops 718 and 720 are clocked by
the signal CLK2, and they provide signals SYNC.sub.-- COE0 and SYNC.sub.--
COE1. The signal SYNC.sub.-- COE0 is provided to one input of a NOR gate 722,
and the signal SYNC.sub.-- COE1 is provided to one input of a NOR gate 724.
The outputs of the NOR gates 722 and 724 are connected to the inputs of
tristate buffers 726 and 728, respectively. The tristate buffers 726 and 728
are enabled by the signal IOBUFOE.sub.-- and drive output enable signals COE0*
and COE1*, respectively. The signals COE0* and COE1* are connected to the
output enable inputs of the first and second banks, respectively, of the cache
SRAMs 209. When asserted low, the signal COEn*, n equal to 0 or 1, enables the
selected SRAMs in bank n to drive data from the modified line onto the data bus
PD.

Detailed Description Text - DETX (107):
Next, the state machine WBST transitions from state F to state A if the signal MORE.sub.-- LINES is low. If MORE.sub.-- LINES is high, the state machine WBST transitions to state G to begin the transfer of the next line.
The transition from state F to state A causes the signal L2WB.sub.-- HOLD.sub.-- CA to be deasserted low to allow a new address from the processor address bus PA to flow into asynchronous cache SRAMs 209. Bits L2WB.sub.-- WORD[1:0] are incremented to the value 0b11 to indicate the fourth and last data transfer in the burst write back cycle. The signal L2WB.sub.-- QP2MWB is maintained high, and a signal L2WB.sub.-- DONE.sub.-- DLY is asserted high.
The signal L2WB.sub.-- DONE.sub.-- DLY is asserted high to indicate that the writeback cycle to the main memory 214 has been completed. In the transition from state F to state A, if the signal GREENMODE is asserted high indicating that the microprocessor 200 is in either the stop grant state or the auto halt power down state, then a signal D.sub.-- L2WB.sub.-- GREEN.sub.-- CADS is asserted high. Referring back to FIG. 5, the signal D.sub.-- L2WB.sub.-- GREEN.sub.-- CADS is provided to inputs of the OR gates 502, 504 and 506, which control the state of the signals SYNC.sub.-- CADS, SYNC.sub.-- BLK.sub.-- CCS0, and SYNC.sub.-- BLK.sub.-- CCS1, respectively. Thus, assertion of the signal D.sub.-- L2WB.sub.-- GREEN.sub.-- CADS places the cache SRAMs 209 back into **low power mode**.

Detailed Description Text - DETX (108):
Thus, a circuit has been described for placing an external or L2 cache memory into **low power mode** in response to certain special cycles executed by the microprocessor. In particular, the special cycles are the stop grant acknowledge special cycle and the halt special cycle. The microprocessor executes the stop grant acknowledge special cycle in response to a request by the computer system to slow down its clock. This request is asserted by the computer system if the system has been idle for a predetermined period

of time.
The halt special cycle is generated by the microprocessor when a HLT instruction is executed. The stop grant acknowledge and halt special cycles
place the microprocessor into a low power state. Since the microprocessor is
in **low power mode,** the L2 cache memory is also placed into **low power mode** for
further power conservation. The L2 cache memory is implemented either with
synchronous or asynchronous static random access memories (SRAMs). To place a
synchronous SRAM into **low power mode,** its address strobe input is asserted
while its chip select input is deasserted. For an asynchronous SRAM, deasserting its chip select input causes the SRAM to transition into **low power
mode**.

Claims Text - CLTX (12):
a **snoop** cycle generator coupled to said bus for generating a **snoop** cycle
having a **snoop address on said bus, wherein said cache** memory device further
includes a second address strobe input, wherein said microprocessor includes an
internal **cache, wherein said microprocessor is able to respond to said snoop**
cycle while said microprocessor is in its low power consumption mode, wherein
said microprocessor generates a writeback cycle to said main memory if said
snoop address corresponds to a modified location in said internal cache,
wherein said microprocessor asserts a processor address strobe signal to
initiate said writeback cycle, said processor address strobe signal being
provided to said second address strobe input of said cache memory device,
wherein asserting said processor address strobe signal causes said cache memory
device to come out of low power state if said cache memory device was previously in low power state, and wherein said circuit further includes:

Claims Text - CLTX (17):
3. The computer system of claim 2, wherein said computer system further
includes a stop **clock generator** for asserting a stop clock signal if said
computer system has been idle for a predetermined period of time, wherein said
microprocessor includes a stop clock input for receiving said stop

clock
signal, wherein said microprocessor further includes a clock input, said stop
clock signal being a request to slow down or stop said clock input to place
said microprocessor into low power consumption mode, and wherein said
microprocessor generates said stop grant acknowledge cycle on said bus in
response to assertion of said stop clock signal.

Claims Text - CLTX (31):
a **snoop** cycle generator coupled to said bus for generating a **snoop** cycle
having a **snoop** address on said bus;

Claims Text - CLTX (32):
a **snoop** address decoder coupled to said bus for determining if said **snoop**
address is in said cache memory device;

Claims Text - CLTX (33):
a modified location detector coupled to said **snoop** address decoder for
asserting a second signal if said **snoop** address corresponds to a modified
location in said cache memory device; and

Claims Text - CLTX (40):
a **snoop** cycle generator coupled to said bus for generating a **snoop** cycle
having a **snoop address on said bus, wherein said cache** memory device includes
an address strobe input, wherein said microprocessor includes an internal
cache, wherein said microprocessor is able to respond to said snoop cycle while
said microprocessor is in its low power consumption mode, wherein said
microprocessor generates a writeback cycle to said main memory if said **snoop**
address corresponds to a modified location in said internal cache, wherein said
chip select signal providing means is further responsive to said writeback
cycle, and chip select signal being asserted if said writeback cycle is active,
wherein asserting said chip select signal causes said cache memory device to
transition out of low power state if said cache memory device was previously in
low power state, and wherein said circuit further includes:

Claims Text - CLTX (43):
a **cache snoop** address detector coupled to said bus for determining if said
snoop address is in said cache memory device;

Claims Text - CLTX (44):
a comparator coupled to said **cache snoop** address detector for asserting a
second signal if said **snoop** address corresponds to a modified location in said
cache memory device; and